

CLAIMS

1. A method of recognising command related items in a  
body of object code, said command related items  
5 corresponding to command names and/or associated option  
names from a textual programming language;

the method comprising:

10 entering a list of entries each comprising a required  
command name and/or option names in programming language  
textual form into a filter table;

15 scanning the body of object code for all bit strings  
potentially representing command names and identifying such  
command names;

20 for each potential command name so identified,  
examining a number of succeeding bits for bits which  
represent valid options for each said command name to  
further identify commands having valid combinations of  
command names and options; and

25 for said identified commands , comparing said  
identified command names and/or option names in programming  
language textual form with the entries of said filter table

to determine whether or not they match any of the list of required command names and/or options in said filter table.

2. A method as claimed in claim 1 including the further step, after said scanning and examining steps, of validating the syntax of each command comprising a command name followed by one or more valid option names and comparing only validated command names and /or option names with the entries in the filter table.

3. A method as claimed in claim 2 in which said step of syntax validation comprises applying each said command to a syntax tree.

4. A method as claimed in claim 1 in which said filter table entries can specify both the presence and the absence of respective command names and/or option names in the scanned and examined object code.

5. A method as claimed in claim 1 in which at least some of said filter table entries include combinations of command and/or option names, comprising the further step of checking the syntax of said combination entries to the filter table.

6. A method as claimed in claim 1 in which the scanning and examining steps involve comparing object code bit strings with bit strings extracted from a library which represent all possible command names and options for said programming language.

7. An object code recognition system for recognising command related items in a body of object code, said command related items corresponding to command names and/or associated option names from a textual programming language;

the system comprising :

a filter table for holding a list of entries each comprising a required command name and/or option names in programming language form;

an object code scanner for scanning the body of object code for all bit strings potentially representing command names and identifying such command names, said scanner being arranged, in response to identification of each potential command name, to examine a number of succeeding bits for bits which represent valid options for each said command name to further identify commands having valid combinations of command names and options; and

5 a filter for comparing said identified command names and/or option names in programming language textual form with the entries of said filter table to determine whether or not they match any of the list of required command names and/or options in said filter table.

10 8. A system as claimed in claim 7 includes a syntax checker for validating the syntax of each command, comprising a command name followed by one or more valid option names whereby only validated command names and/or option names are compared with the entries in the filter table by said filter.

15 9. A system as claimed in claim 8 wherein said syntax checker includes a syntax tree.

20 10. A system as claimed in claim 7 in which said filter table entries can each specify both the presence and the absence of respective command names and /or option names in the scanned and examined object code and said filter is responsive to said specification in determining whether or not said identified command names and/or option names match said filter table entries

11. A system as claimed in claim 7 in which at least some of said filter table entries include combinations of command and/or option names, further comprising a means for checking the syntax of said combination entries to the filter table.

12. A system as claimed in claim 7 further comprising a library containing bit strings representing all possible command names and options for said programming language.

13. A system as claimed in claim 7 in which said filter is arranged to generate a list of matching commands.

14. A computer program recorded on a medium and executable on a computer to recognise command related items in a body of object code, said command related items corresponding to command names and/or associated option names from a textual programming language;

the program comprising :

a filter table data structure for holding a list of entries each comprising a required command name and/or option names in programming language form;

object code scanner code for scanning the body of object code for all bit strings potentially representing command names and identifying such command names, said scanner code being arranged, in response to identification of each potential command name, to examine a number of succeeding bits for bits which represent valid options for each said command name to further identify commands having valid combinations of command names and options; and

filter code for comparing said identified command names and/or option names in programming language textual form with the entries of said filter table to determine whether or not they match any of the list of required command names and/or options in said filter table.

15. A computer program as claimed in claim 14 in which said object code scanner code includes verb objects for representing and identifying command names;

a parameter decoder object for decoding succeeding bits as potentially valid options on identified commands; and

a syntax object for validating the syntax of each command comprising an identified command name followed by one or more valid option names.

16. A computer program as claimed in claim 15 further including a two dimensional array data structure the rows and columns of which are indexed by each of a pair of supplied bytes in said object code respectively, and a file parser object for supplying successive pairs of object code bytes to said array, the array comprising pointers to respective verb objects for each pair of supplied bytes representing a potentially valid command, the file parser object initiating respective verb objects in response to the return of a pointer from said array.

5  
10  
T 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100